REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
	Technical Report	-

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Homogeneous Self-Dual Algorithms for Stochastic Semidefinite Programming	W911NF-08-1-0530
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
	611102
6. AUTHORS	5d. PROJECT NUMBER
Shengping Jin, K. A. Ariyawansa, Yuntao Zhu	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES	8. PERFORMING ORGANIZATION REPORT NUMBER
Washington State University Office Of Grant & Research Development French Administration Bldg. Pullman, WA 99163 -	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	ARO
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
	50556-MA.4

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

Ariyawansa and Zhu [3] have proposed a new class of optimization problems termed stochastic semidefinite programs (SSDPs) to handle data uncertainty in applications leading to (deterministic) semidefinite programs (DSDPs). For the case where the event space of the random variables in an SSDP is finite they have also derived a classof volumetric barrier decomposition algorithms, and proved polynomial complexityof the short-step and long-step members of the class [2]. When the event space of therandom variables in an SSDP is

15. SUBJECT TERMS

Semidefinite programming, homogeneous self-dual algorithms, computational complexity, stochastic semidefinite programming

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UU	UU	UU	UU		K. Ariyawansa
					19b. TELEPHONE NUMBER
					509-335-3152

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

Report Title

Homogeneous Self-Dual Algorithms for Stochastic Semidefinite Programming

ABSTRACT

Ariyawansa and Zhu [3] have proposed a new class of optimization problems termed stochastic semidefinite programs (SSDPs) to handle data uncertainty in applications leading to (deterministic) semidefinite programs (DSDPs). For the case where the event space of the random variables in an SSDP is finite they have also derived a classof volumetric barrier decomposition algorithms, and proved polynomial complexityof the short-step and long-step members of the class [2]. When the event space of therandom variables in an SSDP is finite, the SSDP is equivalent to a large scale DSDPwith special structure. Polynomial homogeneous self-dual algorithms [11] are animportant class of algorithms that have been introduced for solving (general) DSDPs.It is therefore possible to solve SSDPs by applying homogeneous self-dual algorithmsto their DSDP equivalents. However, such algorithms, while polynomial, will still have high computational complexities in comparison to decomposition algorithms.

In this paper, we show how the special structure in DSDP equivalents of SSDPs can be exploited to design homogeneous self-dual algorithms with computational complexities similar to those of volumetric barrier decomposition algorithms.

Homogeneous Self-Dual Algorithms for Stochastic Semidefinite Programming

Shengping Jin* • K. A. Ariyawansa† • Yuntao Zhu‡

Abstract

Ariyawansa and Zhu [3] have proposed a new class of optimization problems termed stochastic semidefinite programs (SSDPs) to handle data uncertainty in applications leading to (deterministic) semidefinite programs (DSDPs). For the case where the event space of the random variables in an SSDP is finite they have also derived a class of volumetric barrier decomposition algorithms, and proved polynomial complexity of the short-step and long-step members of the class [2]. When the event space of the random variables in an SSDP is finite, the SSDP is equivalent to a large scale DSDP with special structure. Polynomial homogeneous self-dual algorithms [11] are an important class of algorithms that have been introduced for solving (general) DSDPs. It is therefore possible to solve SSDPs by applying homogeneous self-dual algorithms to their DSDP equivalents. However, such algorithms, while polynomial, will still have high computational complexities in comparison to decomposition algorithms. In this paper, we show how the special structure in DSDP equivalents of SSDPs can be exploited to design homogeneous self-dual algorithms with computational complexities similar to those of volumetric barrier decomposition algorithms.

Keywords: Semidefinite programming, homogeneous self-dual algorithms, computational complexity, stochastic semidefinite programming.

1 Introduction

Stochastic programming models have been very useful in dealing with data uncertainty in many applications. Stochastic linear programs were introduced in the 1950s as a paradigm for dealing with uncertainty associated with data in applications leading to linear programs. Since then they have been studied extensively [6, 8, 18, 19].

Deterministic semidefinite programs (DSDPs) have been the focus of intense research during the past 20 years, especially in the context of interior point methods for opti-

*Department of Statistics, Wuhan University of Technology, Wuhan, 430063, China. (`spjin@mail.whut.edu.cn`). The work of this author was conducted when he was visiting Washington State University.

†Department of Mathematics, Washington State University, Pullman, WA 99164-3113, USA. (`ari@wsu.edu`). The work of this author was supported in part by the U.S. Army Research Office under Grant DAAD 19-00-1-0465 and by Award W911NF-08-1-0530.

‡Division of Mathematical and Natural Sciences, Arizona State University, Phoenix, AZ 85069-7100, USA. (`yuntao.zhu@asu.edu`). The work of this author was supported in part by the ASU West MGIA Grant 2007.

mization [1, 10, 13, 16]. DSDPs generalize linear programs and have a wide variety of applications, especially beyond those covered by linear programs.

More recently, Ariyawansa and Zhu [3] (see also [9]) introduced a new model termed *Stochastic Semidefinite Programs* (SSDPs) to handle data uncertainty in applications leading to DSDPs. In this new model, they combine stochastic programming and semidefinite programming together so that the uncertainty in data leading to a semidefinite program can be dealt with in the same way that a stochastic linear program handles uncertainty in data in an application leading to a linear program. At the same time, since semidefinite programs generalize linear programs, stochastic semidefinite programs are also a generalization of stochastic linear programs. See [21] for a preliminary collection of applications of SSDPs.

Ariyawansa and Zhu [2] (see also [9]) have presented a class of volumetric barrier decomposition algorithms for SSDPs and proved polynomial complexity of the short-step and long-step members of the class. Their derivation is for the case where the event space of the random variables in the SSDP is finite. The computational complexity in terms of the number of arithmetic operations of the short-step and long-step algorithms in [2] are $O(K^{1.5})$ and $O(K^2)$ respectively, where K is the number of realizations of the random variables. Another important feature of the algorithms in [2] is that the most expensive part of the algorithm naturally separates into K subproblems, which allows efficient parallel implementations.

When the event space of the random variables in an SSDP is finite, the SSDP can be converted to an equivalent large scale DSDP. Thus such an SSDP can also be solved by using algorithms for DSDPs on the DSDP equivalent to the SSDP. One such general purpose DSDP algorithms is the homogeneous self-dual algorithm (see [11]). However, the computational complexity in terms of the number of arithmetic operations of the general-purpose homogeneous self-dual algorithm on the DSDP equivalent of an SSDP is $O(K^{4.5})$. In the context of SSDPs, K is large, and so $O(K^{4.5})$ is significantly larger than $O(K^{1.5})$ and $O(K^2)$, the complexities of the short-step and long-step algorithms respectively in [2].

In this paper, we propose homogeneous self-dual algorithms for SSDPs (with finite event spaces of their random variables) that exploit the special structure of the equivalent DSDP to reduce the computational complexity to $O(K^{1.5})$. In addition, the most expensive part of the algorithm separates into K subproblems.

In general, algorithms for stochastic optimization belong to two broad categories both exploiting their special structure but in two different ways. First, we have algorithms based on notions of cutting planes and decomposition. The classical L-shaped algorithm in [17], the volumetric center cutting plane algorithm in [4], and the decomposition algorithms in [20], [9] and [2] belong to this category. Second, we have algorithms designed by tailoring the operations in general purpose algorithms to exploit the special structure in stochastic optimization problems. The algorithms in [12, 7, 5] belong to this category. The algorithms derived in this paper for SSDPs also belong to this second category.

The rest of the paper is organized as follows. In the next section we introduce our notation and present some preliminary material on the homogeneous self-dual methods for solving DSDPs. We present our homogeneous self-dual algorithms for SSDPs in §3.

In §4 we present an efficient method that exploits the structure of the DSDP equivalent of an SSDP for computing search directions in homogeneous self-dual methods. We also obtain an estimate of the number of arithmetic operations per iteration of our algorithm in §4. We conclude the paper in §5 where we compare the computational complexities of the new algorithms presented in this paper with those of the algorithms in [2].

2 Preliminaries

2.1 Notation and terminology

The notation and terminology we use in the rest of this paper follow that in [13]. We let $\mathbb{R}^{m\times n}$ and $\mathbb{R}^{n\vee n}$ denote the vector spaces of real $m\times n$ matrices and real symmetric $n\times n$ matrices respectively. We write $X \succeq 0$ ($X \succ 0$) to mean that X is positive semidefinite (positive definite) and we use $U \succeq V$ or $V \preceq U$ to mean that $U - V \succeq 0$. For $U, V \in \mathbb{R}^{m\times n}$ we write $U \bullet V := \operatorname{trace}(U^{\mathsf{T}} V)$ to denote the Frobenius inner product of U and V.

Given $A_i \in \mathbb{R}^{n\vee n}$ for $i = 1, 2, \ldots, m$, we define the operator $\mathcal{A} : \mathbb{R}^{n\vee n} \longrightarrow \mathbb{R}^m$ by

$$\mathcal{A}X = \begin{bmatrix} A_1 \bullet X \\ A_2 \bullet X \\ \vdots \\ A_m \bullet X \end{bmatrix}, \tag{1}$$

for any $X \in \mathbb{R}^{n\vee n}$.

The adjoint operator of $\mathcal{A}$ with respect to the standard inner products in $\mathbb{R}^{n\vee n}$ and $\mathbb{R}^m$ is the operator $\mathcal{A}^* : \mathbb{R}^m \longrightarrow \mathbb{R}^{n\vee n}$ defined by

$$\mathcal{A}^* y = \sum_{i=1}^{m} y_i A_i, \tag{2}$$

for any $y \in \mathbb{R}^m$.

We introduce the useful notation Kronecker product $P \circledast Q$, where P and Q are usually symmetric. This is an operator from $\mathbb{R}^{n\vee n}$ to itself defined by

$$(P \circledast Q)U = \frac{1}{2}(PUQ^{\mathsf{T}} + QUP^{\mathsf{T}}), \tag{3}$$

for any $U \in \mathbb{R}^{n\vee n}$.

Finally we introduce the operator *svec* which transforms a matrix in $\mathbb{R}^{n\vee n}$ to a vector in $\mathbb{R}^{\bar{n}}$ (see [14]) where $\bar{n} := \frac{1}{2}n(n+1)$.

2.2 Homogeneous Self-Dual Methods for DSDPs

Given data $A_i \in \mathbb{R}^{n\vee n}$ for $i = 1, 2, \ldots, m$, $b \in \mathbb{R}^m$ and $C \in \mathbb{R}^{n\vee n}$, a DSDP in *primal standard form* is defined as

$$\begin{array}{rrcl} \text{minimize} & C \bullet X & & \\ \text{subject to} & \mathcal{A}X & = & b \\ & X & \succeq & 0, \end{array} \tag{4}$$

where $X \in \mathbb{R}^{n\vee n}$ is the variable. The dual of (4) is

$$\begin{array}{lrcl} \text{maximize} & b^\mathsf{T} y & & \\ \text{subject to} & \mathcal{A}^* y + S & = & C \\ & S & \succeq & 0, \end{array} \tag{5}$$

where $y \in \mathbb{R}^m$ and $S \in \mathbb{R}^{n\vee n}$ are the variables.

We briefly review the homogeneous interior point algorithm for DSDPs as in [11]. The homogeneous model for (4, 5) is as follows:

$$\begin{array}{rrrrrcl} \mathcal{A}X & & & -b\tau & & = & 0 \\ & -\mathcal{A}^* y & -S & +\tau C & & = & 0 \\ -C \bullet X & +b^\mathsf{T} y & & & -\kappa & = & 0. \end{array} \tag{6}$$

It is easy to show from (6) that

$$X \bullet S \quad + \quad \tau\kappa \quad = \quad 0.$$

The main step at each iteration of the homogeneous interior point algorithm (shown below in Algorithm 1) is the computation of the search direction $(\Delta X, \Delta y, \Delta Z)$ from the symmetrized Newton equations with respect to an invertible matrix P (which is chosen as a function of (X, y, S)) defined by:

$$\begin{array}{rrrrrcl} \mathcal{A}\Delta X & & & -b\Delta\tau & & = & \eta r_p \\ & -\mathcal{A}^*\Delta y & -\Delta S & +\Delta\tau C & & = & \eta R_d \\ -C \bullet \Delta X & +b^\mathsf{T}\Delta y & & & -\Delta\kappa & = & \eta r_g \\ & & & \kappa\Delta\tau & +\tau\Delta\kappa & = & \gamma\mu - \tau\kappa \\ \mathcal{E}\Delta X & & +\mathcal{F}\Delta S & & & = & \gamma\mu I - H_P(XS) \end{array} \tag{7}$$

where $r_p := b\tau - \mathcal{A}X$, $R_d := \mathcal{A}^* y + S - \tau C$, $r_g := C \bullet X - b^\mathsf{T} y + \kappa$, $\mu := [1/(n+1)](X \bullet S + \tau\kappa)$, η and γ are two parameters, $H_P : \mathbb{R}^{n\vee n} \longrightarrow \mathbb{R}^{n\vee n}$ is the symmetrization operator defined by

$$H_P(U) := \frac{1}{2}(PUP^{-1} + (P^{-1})^\mathsf{T} U^\mathsf{T} P^\mathsf{T}),$$

and $\mathcal{E} : \mathbb{R}^{n\vee n} \longrightarrow \mathbb{R}^{n\vee n}$ and $\mathcal{F} : \mathbb{R}^{n\vee n} \longrightarrow \mathbb{R}^{n\vee n}$ are the linear operators defined by

$$\mathcal{E} := P \circledast (P^{-1})^\mathsf{T} S; \quad \mathcal{F} := PX \circledast (P^{-1})^\mathsf{T},$$

respectively.

Currently, the most common choices of symmetrization for search directions in practice are as follows [15].

1. Helmberg-Rendel-Vanderbei-Wolkowicz/Kojima-Shindoh-Hara/Monteiro (HKM) direction, corresponding to $P := S^{1/2}$. In this case, we have that $\mathcal{E} = \mathcal{I}$ and $\mathcal{F} = X \circledast S^{-1}$.

2. Kojima, Shindoh, and Hara (KSH) direction (rediscovered by Monteiro), corresponding to $P := X^{-1/2}$. In this case, we have that $\mathcal{E} = S \circledast X^{-1}$ and $\mathcal{F} = \mathcal{I}$.

3. Nesterov-Todd (NT) direction, corresponding to $P := H^{-1/2}$, here H is the unique symmetric positive definite matrix satisfying $HXH = X$, which can be calculated by

$$H = X^{1/2}\left(X^{1/2}SX^{1/2}\right)^{-1/2}X^{1/2}.$$

In this case, we have $\mathcal{E} = \mathcal{I}$ and $\mathcal{F} = H \circledast H$.

Lemma 1. *Suppose that $X \succ 0$, $S \succ 0$, and $A_1, A_2, \ldots, A_m$ are linearly independent. Then for each of the above three choices of P, $\mathcal{E}^{-1}$ and $\mathcal{F}^{-1}$ exist, and $\mathcal{E}^{-1}\mathcal{F}$ and $\mathcal{F}^{-1}\mathcal{E}$ are positive definite and self-adjoint.*

Proof. See [13]. □

We state the generic homogeneous algorithm as in [11].

Algorithm 1 Generic Homogeneous Self-Dual Algorithm for Solving (4, 5)

$(X, y, S, \tau, \kappa) := (I, 0, I, 1, 1)$
while a stopping criterion is not satisfied **do**
  choose η, γ
  compute the solution $(\Delta X, \Delta y, \Delta S, \Delta\tau, \Delta\kappa)$ of the linear system (7)
  compute a step length $\bar\theta$ so that
  $X + \bar\theta\Delta X \succ 0$,
  $S + \bar\theta\Delta S \succ 0$,
  $\tau + \bar\theta\Delta\tau > 0$, and
  $\kappa + \bar\theta\Delta\kappa > 0$
  $(X, y, S, \tau, \kappa) := (X, y, S, \tau, \kappa) + \bar\theta(\Delta X, \Delta y, \Delta S, \Delta\tau, \Delta\kappa)$
end while

3 Homogeneous Self-Dual Algorithms for SSDPs

3.1 Definition of a SSDP in Primal Standard Form

As in [3], we define a SSDP with recourse in primal standard form based on deterministic data $W_{0_i} \in \mathbb{R}^{n_0 \vee n_0}$ for $i = 1, 2, \ldots, m_0$, $h_0 \in \mathbb{R}^{m_0}$ and $C_0 \in \mathbb{R}^{n_0 \vee n_0}$; and random data $B_i(\omega) \in \mathbb{R}^{n_0 \vee n_0}$, $W_i(\omega) \in \mathbb{R}^{n_1 \vee n_1}$ for $i = 1, 2, \ldots, m_1$, $h(\omega) \in \mathbb{R}^{m_1}$ and $C(\omega) \in \mathbb{R}^{n_1 \vee n_1}$ that depend on an underlying outcome ω in an event space Ω with a known probability function $\mathbb{P}$. Given this data, a SSDP with recourse in primal standard form is

$$\begin{array}{rrcl} \text{minimize} & C_0 \bullet X_0 & + & \mathbb{E}\left[Q\left(X_0, \omega\right)\right] \\ \text{subject to} & \mathcal{W}_0 X_0 & = & h_0 \\ & X_0 & \succeq & 0, \end{array} \tag{8}$$

where $X_0 \in \mathbb{R}^{n_0 \vee n_0}$ is the first-stage decision variable, $Q\left(X_0, \omega\right)$ is the minimum of the problem

$$\begin{array}{rccccl} \text{minimize} & C(\omega) \bullet Y(\omega) & & & & \\ \text{subject to} & \mathcal{B}(\omega) X_0 & + & \mathcal{W}(\omega) Y(\omega) & = & h(\omega) \\ & & & Y(\omega) & \succeq & 0, \end{array} \tag{9}$$

where $Y(\omega) \in \mathbb{R}^{n_1 \vee n_1}$ is the second-stage variable, and

$$\mathbb{E}\left[Q\left(X_0, \omega\right)\right] = \int_{\Omega} Q\left(X_0, \omega\right) \mathbb{P}(d\omega). \tag{10}$$

When the event space Ω is finite, Problem (8, 9, 10) is equivalent to a DSDP. The equivalent DSDP has a special block angular structure which is exploited to reduce the computational complexity.

3.2 Solving SSDPs with Finite Event Space

We consider SSDP (8, 9, 10) when the event space Ω is finite. Let $\{(C_k, \mathcal{B}_k, \mathcal{W}_k, h_k), k = 1, 2, \ldots, K\}$ be the possible values of the random variables $(C(\omega), \mathcal{B}(\omega), \mathcal{W}(\omega), h(\omega))$ and let p_k be the corresponding probabilities for $k = 1, 2, \ldots, K$. For convenience, let $C_k := p_k C_k$ for $k = 1, 2, \ldots, K$. Then Problem (8, 9, 10) is equivalent to

$$\begin{array}{rccccccccc}
\text{minimize} & C_0 \bullet X_0 & + & C_1 \bullet X_1 & + & \cdots & + & C_K \bullet X_K & & \\
\text{subject to} & \mathcal{W}_0 X_0 & & & & & & & = & h_0 \\
& \mathcal{B}_1 X_0 & + & \mathcal{W}_1 X_1 & & & & & = & h_1 \\
& \vdots & & & & \ddots & & & & \vdots \\
& \mathcal{B}_K X_0 & & & & & + & \mathcal{W}_K X_K & = & h_K \\
& X_0, & & X_1, & & \cdots & & X_K & \succeq & 0
\end{array} \tag{11}$$

where $X_0 \in \mathbb{R}^{n_0 \vee n_0}$ and $X_k \in \mathbb{R}^{n_1 \vee n_1}$ for $k = 1, 2, \ldots, K$ are the first-stage and second-stage variables respectively.

Problem (11) is a DSDP in primal standard form. To see this, first let

$$C := diag(C_0, C_1, \ldots, C_K) \in \mathbb{R}^{(n_0 + K n_1) \vee (n_0 + K n_1)},$$

$$X := diag(X_0, X_1, \ldots, X_K) \in \mathbb{R}^{(n_0 + K n_1) \vee (n_0 + K n_1)},$$

and

$$h^{\mathsf{T}} := \left[h_0^{\mathsf{T}}, h_1^{\mathsf{T}}, \ldots, h_K^{\mathsf{T}}\right] \in \mathbb{R}^{m_0 + K m_1}.$$

Next for $i = 1, 2, \ldots, (m_0 + K m_1)$, define $A_i \in \mathbb{R}^{(n_0 + K n_1) \vee (n_0 + K n_1)}$ by

$$\begin{array}{ll}
A_i := diag(W_{0_i}, 0, \ldots, 0), & i = 1, 2, \ldots, m_0 \\
A_{m_0 + i} := diag(B_{1_i}, W_{1_i}, 0, \ldots, 0), & i = 1, 2, \ldots, m_1 \\
\vdots & \vdots \\
A_{m_0 + (K-1) m_1 + i} := diag(B_{K_i}, 0, \ldots, 0, W_{K_i}), & i = 1, 2, \ldots, m_1.
\end{array}$$

Then (11) becomes

$$\begin{array}{rrcll}
\text{minimize} & C \bullet X & & & \\
\text{subject to} & A_i \bullet X & = & h_i, & i = 1, 2, \ldots, (m_0 + K m_1) \\
& X & \succeq & 0. &
\end{array} \tag{12}$$

Problem (12) is same as (4) if we set $n := (n_0 + K n_1)$ and $m := (m_0 + K m_1)$. So Problem (11) is a DSDP in primal standard form with block diagonal structure.

Algorithms that exploit this special structure is especially important when K is large as is the case in typical applications.

The dual of (11) is

$$
\begin{array}{rccccccccccc}
\text{maximize} & h_0^\mathsf{T} y_0 & + & h_1^\mathsf{T} y_1 & + & \cdots & + & h_K^\mathsf{T} y_K & & & & \\
\text{subject to} & \mathcal{W}^* y_0 & + & \mathcal{B}_1^* y_1 & + & \cdots & + & \mathcal{B}_K^* y_K & + & S_0 & = & C_0 \\
 & & & \mathcal{W}_1^* y_1 & & & & & + & S_1 & = & C_1 \\
 & & & & & \ddots & & & & & & \vdots \\
 & & & & & & & \mathcal{W}_K^* y_K & + & S_K & = & C_K \\
 & S_0, & & S_1, & & \cdots & & S_K & & & \succeq & 0,
\end{array}
\tag{13}
$$

where $y \in \mathbb{R}^{(m_0+Km_1)}$ and $S_k \in \mathbb{R}^{(n_0+Kn_1)\vee(n_0+Kn_1)}$ for $k = 1, 2, \ldots, K$ are the variables.

Now the homogeneous interior point method can be applied to Problem (11, 13). However, the size of Problem (11, 13) increases as K increases. In practice K is typically very large. So in practice, the Problem (11, 13) is large-scale and the computation of the search direction in Algorithm 1 (i.e. the solution of the system 7) is very expensive. As we shall see in the next section, this computational work can be reduced significantly by exploiting the special structure of Problem (11, 13). The method we describe in the next section for the computation of the search direction has an additional desirable feature: it decomposes into K smaller computations that can be performed in parallel.

4 An Efficient Method for Computing Search Directions

We now describe a method for computing the search direction in Algorithm 1 that exploits the special structure in (11, 13).

The homogeneous model (6) for Problem(11, 13) is

$$
\begin{array}{lcl}
\mathcal{W}_0 X_0 - h_0 \tau & = & 0 \\
\mathcal{B}_k X_0 + \mathcal{W}_k X_k - h_k \tau & = & 0, \ k = 1, 2, \ldots, K \\
-\mathcal{W}_0^* y_0 - \sum_{k=1}^K \mathcal{B}_k^* y_k + \tau C_0 - S_0 & = & 0 \\
-\mathcal{W}_k^* y_k + \tau C_k - S_k & = & 0, \ k = 1, 2, \ldots, K \\
\sum_{k=0}^K h_k^\mathsf{T} y_k - \sum_{k=0}^K C_k \bullet X_k - \kappa & = & 0 \\
X_k \succeq 0, \ S_k \succeq 0, \ k = 0, 1, 2, \ldots, K, \ \tau \geq 0, \ \kappa \geq 0, & &
\end{array}
\tag{14}
$$

where $\mathcal{W}_k^*$ and $\mathcal{B}_k^*$ for $k = 1, 2, \ldots, K$ are adjoint operators in the sense of (2) with appropriate dimensions.

The search direction system corresponding to (14) can be derived via (7) as

$$
\begin{array}{lcl}
\mathcal{W}_0 \Delta X_0 - h_0 \Delta\tau & = & \eta r_{p_0} \\
\mathcal{B}_k \Delta X_0 + \mathcal{W}_k \Delta X_k - h_k \Delta\tau & = & \eta r_{p_k}, \ k = 1, 2, \ldots, K \\
-\mathcal{W}_0^* \Delta y_0 - \sum_{k=1}^K \mathcal{B}_k^* \Delta y_k + & & \\
\qquad \Delta\tau C_0 - \Delta S_0 & = & \eta R_{d_0} \\
-\mathcal{W}_k^* \Delta y_k + \Delta\tau C_k - \Delta S_k & = & \eta R_{d_k}, \ k = 1, 2, \ldots, K \\
\mathcal{E}_0 \Delta X_0 + \mathcal{F}_0 \Delta S_0 & = & \gamma\mu I_0 - H_{P_0}(X_0 S_0) \\
\mathcal{E}_k \Delta X_k + \mathcal{F}_k \Delta S_k & = & \gamma\mu I_k - H_{P_k}(X_k S_k), \ k = 1, 2, \ldots, K \\
\kappa \Delta\tau + \tau \Delta\kappa & = & \gamma\mu - \tau\kappa \\
\sum_{k=0}^K h_k^\mathsf{T} \Delta y_k - \sum_{k=0}^K C_k \bullet \Delta X_k - \Delta\kappa & = & \eta r_g,
\end{array}
\tag{15}
$$

where

$$\begin{array}{rcl} r_{p_0} & = & h_0\tau - \mathcal{W}_0 X_0 \\ r_{p_k} & = & h_k\tau - \mathcal{B}_k X_0 - \mathcal{W}_k X_k,\ k = 1, 2, \ldots, K \\ R_{d_0} & = & \mathcal{W}_0^* y_0 + \sum_{k=1}^K \mathcal{B}_k^* y_k + S_0 - \tau C_0 \\ R_{d_k} & = & \mathcal{W}_k^* y_k + S_k - \tau C_k,\ k = 1, 2, \ldots, K \\ r_g & = & \kappa - \sum_{k=0}^K h_k^{\mathsf{T}} y_k + \sum_{k=0}^K C_k \bullet X_k \\ \mu & = & \frac{1}{n_0 + K n_1 + 1}(\sum_{k=0}^K X_0 \bullet S_0 + \tau\kappa), \end{array}$$

η and γ are two parameters, and $\mathcal{E}_k, \mathcal{F}_k$ and H_{P_k} are linear operators which depend only on X_k and S_k.

Now we present the crux of our method for finding the search direction as a solution to (15). By the sixth equation of (15) and the fact that $\mathcal{F}_k^{-1}$ for $k = 1, 2, \ldots, K$ exist, we have

$$\Delta S_k \quad = \quad -\mathcal{F}_k^{-1}\mathcal{E}_k \Delta X_k + \mathcal{F}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_k S_k)), \quad k = 1, 2, \ldots, K. \tag{16}$$

Substituting (16) into the fourth equation of (15), we get

$$-\mathcal{W}_k^* \Delta y_k + \Delta\tau C_k + \mathcal{F}_k^{-1}\mathcal{E}_k \Delta X_k = \eta R_{d_k} + \mathcal{F}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_k S_k)) \tag{17}$$

for $k = 1, 2, \ldots, K$. Since $\mathcal{E}_k^{-1}$ exist, we have $(\mathcal{F}_k^{-1}\mathcal{E}_k)^{-1} = \mathcal{E}_k^{-1}\mathcal{F}_k$, for $k = 1, 2, \ldots, K$. By equation (17), for $k = 1, 2, \ldots, K$, ΔX_k can be expressed as

$$\Delta X_k = \mathcal{E}_k^{-1}\mathcal{F}_k \mathcal{W}_k^* \Delta y_k - \mathcal{E}_k^{-1}\mathcal{F}_k(\Delta\tau C_k + \eta R_{d_k}) + \mathcal{E}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_k S_k)). \tag{18}$$

Substituting (18) into the second equation of (15), we have

$$\begin{array}{l} \mathcal{B}_k \Delta X_0 + \mathcal{W}_k(\mathcal{E}_k^{-1}\mathcal{F}_k \mathcal{W}_k^* \Delta y_k - \mathcal{E}_k^{-1}\mathcal{F}_k(\Delta\tau C_k + \eta R_{d_k}) + \mathcal{E}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_k S_k))) \\ \quad -h_k \Delta\tau = \eta r_{p_k}, \quad k = 1, 2, \ldots, K. \end{array} \tag{19}$$

Thus

$$\Delta y_k = M_k^{-1}\mathcal{B}_k \Delta X_0 + q_k \Delta\tau + \nu_k \tag{20}$$

where

$$\begin{array}{l} M_k = \mathcal{W}_k \mathcal{E}_k^{-1}\mathcal{F}_k \mathcal{W}_k^* \\ q_k = M_k^{-1}(\mathcal{W}_k \mathcal{E}_k^{-1}\mathcal{F}_k + h_k) \\ \nu_k = M_k^{-1}(\eta r_{p_k} - \eta \mathcal{W}_k \mathcal{E}_k^{-1}\mathcal{F}_k R_{d_k} - \mathcal{W}_k \mathcal{E}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_k S_k))), \end{array} \tag{21}$$

for $k = 1, 2, \ldots, K$.
By the fifth equation of (15) and the fact that $\mathcal{F}_0^{-1}$ exists, one gets

$$\Delta S_0 = -\mathcal{F}_0^{-1}\mathcal{E}_0 \Delta X_0 + \mathcal{F}_0^{-1}(\gamma\mu I_0 - H_{P_0}(X_0 S_0)). \tag{22}$$

We substitute (20) and (22) in the third equation of (15) to get

$$\begin{array}{l} -\mathcal{W}_0^* \Delta y_0 - \sum_{k=1}^K \mathcal{B}_k^*(-M_k^{-1}\mathcal{B}_k \Delta X_0 + q_k \Delta\tau + \nu_k) + \Delta\tau C_0 + \mathcal{F}_0^{-1}\mathcal{E}_0 \Delta X_0 \\ \quad -\mathcal{F}_0^{-1}(\gamma\mu I_0 - H_{P_0}(X_0 S_0)) = \eta R_{d_0}. \end{array} \tag{23}$$

From (23) we have

$$\begin{array}{rcl} \Delta X_0 & = & \mathcal{M}_0^{-1}(\mathcal{W}_0^* \Delta y_0 + (\sum_{k=1}^K \mathcal{B}_k^* q_k - C_0)\Delta\tau) \\ & & +\mathcal{M}_0^{-1}(\sum_{k=1}^K \mathcal{B}_k^* \nu_k + \eta R_{d_0} + \mathcal{F}_0^{-1}(\gamma\mu I_0 - H_{P_0}(X_0 S_0))) \\ & = & \mathcal{M}_0^{-1}\mathcal{W}_0^* \Delta y_0 - T_0 \Delta\tau + U_0, \end{array} \tag{24}$$

where

$$\begin{array}{rcl} \mathcal{M}_0 & = & \mathcal{F}_0^{-1}\mathcal{E}_0 + \sum_{k=1}^K \mathcal{B}_k^* M_k^{-1} \mathcal{B}_k \\ T_0 & = & \mathcal{M}_0^{-1}(C_0 - \sum_{k=1}^K \mathcal{B}_k^* q_k) \\ U_0 & = & \mathcal{M}_0^{-1}(\sum_{k=1}^K \mathcal{B}_k^* \nu_k + \eta R_{d_0} + \mathcal{F}_0^{-1}(\gamma\mu I_0 - H_{P_0}(X_0 S_0))). \end{array} \tag{25}$$

Now substituting (24) into the first equation of (15), we have

$$\mathcal{W}_0(\mathcal{M}_0^{-1}\mathcal{W}_0^* \Delta y_0 - T_0 + U_0) - h_0 \Delta\tau \ = \ \eta r_{p_0}. \tag{26}$$

From (26) and the fact that $\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*$ is nonsingular (this will be discussed in detail later in §5), we have that

$$\begin{array}{rcl} \Delta y_0 & = & (\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*)^{-1}((\mathcal{W}_0 T_0 + h_0)\Delta\tau + \eta r_{p_0} + \mathcal{W}_0 U_0 \\ & = & \alpha_0 \Delta\tau + \beta_0, \end{array} \tag{27}$$

where

$$\begin{array}{rcl} \alpha_0 & = & (\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*)^{-1}(\mathcal{W}_0 T_0 + h_0) \\ \beta_0 & = & (\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*)^{-1}(\eta r_{p_0} - \mathcal{W}_0 U_0). \end{array} \tag{28}$$

Now we substitute backwards. First we substitute (27) in (24) to get

$$\begin{array}{rcl} \Delta X_0 & = & \mathcal{M}_0^{-1}\mathcal{W}_0^*(\alpha_0\Delta\tau + \beta_0) - T_0\Delta\tau + U_0 \\ & = & \Psi_0 \Delta\tau + \Phi_0, \end{array} \tag{29}$$

where

$$\begin{array}{rcl} \Psi_0 & = & \mathcal{M}_0^{-1}\mathcal{W}_0^* \alpha_0 - T_0 \\ \Phi_0 & = & \mathcal{M}_0^{-1}\mathcal{W}_0^* \beta_0 + U_0. \end{array} \tag{30}$$

Substituting (29) in (20), we obtain

$$\begin{array}{rcl} \Delta y_k & = & -(\mathcal{W}_k \mathcal{E}_k^{-1}\mathcal{F}_k \mathcal{W}_k^*)^{-1}\mathcal{B}_k(\Psi_0\Delta\tau + \Phi_0) + q_k\Delta\tau + \nu_k \\ & = & \alpha_k \Delta\tau + \beta_k, \end{array} \tag{31}$$

where

$$\begin{array}{rcl} \alpha_k & = & -(\mathcal{W}_k \mathcal{E}_k^{-1}\mathcal{F}_k \mathcal{W}_k^*))^{-1}\mathcal{B}_k \Psi_0 + q_k \\ \beta_k & = & -(\mathcal{W}_k \mathcal{E}_k^{-1}\mathcal{F}_k \mathcal{W}_k^*))^{-1}\mathcal{B}_k \Phi_0 + \nu_k \end{array} \tag{32}$$

for $k = 1, 2, \ldots, K$. Furthermore, we substitute (31) in (18) to get

$$\begin{array}{rcl} \Delta X_k & = & \mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^*(\alpha_k\Delta\tau + \beta_k) - \mathcal{E}_k^{-1}\mathcal{F}_k(\Delta\tau C_k + \eta R_{d_k}) + \mathcal{E}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_k S_k)) \\ & = & \Psi_k \Delta\tau + \Phi_k, \quad k = 1, 2, \ldots, K, \end{array} \tag{33}$$

where

$$\begin{array}{rcl} \Psi_k & = & \mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^* \alpha_k - \mathcal{E}_k^{-1}\mathcal{F}_k C_k \\ \Phi_k & = & \mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^* \beta_k - \mathcal{E}_k^{-1}\mathcal{F}_k \eta R_{d_k} + \mathcal{E}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_k S_k)). \end{array} \tag{34}$$

for $k = 1, 2, \ldots, K$. Now, we substitute (27), (29), (31) and (33) in the last equation of (15). By the seventh equation of (15) this yields

$$\sum_{k=0}^{K} h_k^{\mathsf{T}}(\alpha_k \Delta\tau + \beta_k) - \sum_{k=0}^{K} C_k \bullet (\Psi_k \Delta\tau + \Phi_k) - \frac{1}{\tau}(-\kappa\Delta\tau + \gamma\mu - \tau\kappa) = \eta r_g.$$

Finally $\Delta\tau$ is given by

$$\Delta\tau \quad = \quad \frac{\tau\eta r_g + \tau\sum_{k=0}^{K}(C_k \bullet \Phi_k - h_k^{\mathsf{T}}\beta_k) + (\gamma\mu - \tau\kappa)}{\tau\sum_{k=0}^{K}(h_k^{\mathsf{T}}\alpha_k - C_k \bullet \Psi_k) + \kappa}. \tag{35}$$

All the other directions can be obtained by (35). Once all the directions are computed, we can iterate to the next point in Algorithm 1.

5 Complexity Analysis

In this section we first show that under reasonable conditions the operations described above are valid. Then we estimate the computational complexity of Algorithm 1 with the method described in §4 applied on Problem (11, 13). Finally, we compare and compare that complexity to the complexity of Algorithm 1 applied on Problem (11, 13) treating it as a generic primal-dual DSDP pair.

5.1 Validation of Computations

We assume that $W_0^{(1)}, W_0^{(2)}, \ldots, W_0^{(m_0)}$, and $W_k^{(1)}, W_k^{(2)} \ldots, W_k^{(m_1)}$ for $k = 1, 2, \ldots, K$ are linearly independent. Then M_k for $k = 1, 2, \ldots, K$ in 21) are nonsingular and positive definite by Lemma 1.

Now we will show that $\mathcal{M}_0$ in (25) is nonsingular and that $\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*$ is also nonsingular.

Lemma 2. *Suppose that $W_0^{(1)}, W_0^{(2)}, \ldots, W_0^{(m_0)}$ and that $W_k^{(1)}, W_k^{(2)} \ldots, W_k^{(m_1)}$ are linearly independent for $k = 1, 2, \ldots, K$. Then $\mathcal{M}_0$ in (25) and $\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*$ are positive definite.*

Proof. From (25) we have

$$\mathcal{M}_0 \quad = \quad \mathcal{F}_0^{-1}\mathcal{E}_0 + \textstyle\sum_{k=1}^{K} \mathcal{B}_k^* M_k^{-1} \mathcal{B}_k.$$

We have that $\mathcal{F}_0^{-1}\mathcal{E}_0$ is positive definite by Lemma 1, so it suffices to show that $\mathcal{B}_k^* M_k^{-1}\mathcal{B}_k$ is positive semidefinite for $k = 1, 2, \ldots, K$. In fact, denoting

$$\mathcal{B}_k U = \left[B_k^{(1)} \bullet U, B_k^{(2)} \bullet U, \ldots, B_k^{(m_1)} \bullet U\right]^{\mathsf{T}}$$

for each $U \in \mathbb{R}^{n_0 \vee n_0}$ and $M_k^{-1} = \left[\phi_{ij}^{(k)}\right]_{m_1 \times m_1}$, we have

$$\begin{aligned} \mathcal{B}_k^* M_k^{-1} \mathcal{B}_k U \quad &= \quad \mathcal{B}_k^* \left[\phi_{ij}^{(k)}\right]_{m_1 \times m_1} \left[B_k^{(1)} \bullet U, \ldots, B_k^{(m_1)} \bullet U\right]^{\mathsf{T}} \\ &= \quad \textstyle\sum_i \sum_j (\phi_{ij}^{(k)} B_k^{(j)} \bullet U) B_k^{(i)} \end{aligned}$$

for $k = 1, 2, \ldots, K$. So

$$\begin{array}{rcl}\mathcal{B}_k^* M_k^{-1} \mathcal{B}_k U \bullet U & = & \sum_i \sum_j (\phi_{ij}^{(k)} B_k^{(j)} \bullet U) B_k^{(i)} \bullet U \\ & = & (\mathcal{B}_k U)^{\mathsf{T}} M_k^{-1} (\mathcal{B}_k U) \geq 0.\end{array}$$

The last inequality is due to the fact that M_k^{-1} is positive definite. As in [13] we can then show that $\mathcal{W}_0 \mathcal{M}_0^{-1} \mathcal{W}_0^*$ is positive definite. This completes the proof. □

5.2 The Complexity Analysis

Theorem 1. *Suppose that $m_1 = m_0$, $n_1 = n_0$ and that $W_k^{(1)}, W_k^{(2)}, \ldots, W_k^{(m_0)}$ are linearly independent for $k = 0, 1, \ldots, K$. By utilizing method described in §4 for computing the search directions in Algorithm 1, we have that the number of arithmetic operations in each iteration of Algorithm 1 is $O(K(m_0^3 + m_0^2 n_0^4) + n_0^6)$.*

Proof. The dominant computations of the method in §4 occur at (21), (25), (28), (30), (32) and (34). The corresponding numbers of arithmetic operations of these computations are listed in Table 1. In particular, the computation in (25) will be analyzed in detail. The total number of arithmetic operations is dominated by $O(K(m_0^3+m_0^2n_0^4)+n_0^6)$

Table 1: Complexity Estimates for Dominant Steps in Method of §4

Equation Number of Computation	Estimate of the Number of Arithmetic Operations
(21)	$O(K(m_0^3 + m_0 n_0^3))$
(25)	$O(K m_0^2 n_0^4 + n_0^6)$
(28)	$O(m_0^2 n_0^2 + m_0^3)$
(30)	$O(m_0 n_0^2)$
(32)	$O(m_0^2 n_0^2 + m_0^3)$
(34)	$O(m_0 n_0^2)$

for all three choices of P indicated in §2.2.

To analyze the work of computation (25), we let

$$\begin{array}{rcl} svec(\mathcal{B}_k^* M_k^{-1} \mathcal{B}_k U) & = & svec(\sum_i \sum_j (\phi_{ij}^{(k)} B_k^{(j)} \bullet U) B_k^{(i)}) \\ & = & \sum_i \sum_j (\phi_{ij}^{(k)} B_k^{(j)} \bullet U) svec(B_k^{(i)}) \\ & = & \sum_i \sum_j \phi_{ij}^{(k)} (svec(B_k^{(j)})^{\mathsf{T}} svec(U)) svec(B_k^{(i)}) \\ & = & \sum_i \sum_j \phi_{ij}^{(k)} (svec(B_k^{(i)}) svec(B_k^{(j)})^{\mathsf{T}}) svec(U). \end{array}$$

So matrix $\mathcal{M}_0 = \mathcal{F}_0^{-1} \mathcal{E}_0 + \sum_{k=1}^K \mathcal{B}_k^* M_k^{-1} \mathcal{B}_k$ in $\mathbb{R}^{n_0 \vee n_0}$ is given by

$$H_0 + \sum_{k=1}^{K} \sum_{i=1}^{m_0} \sum_{j=1}^{m_0} \phi_{ij}^{(k)} (svec(B_k^{(i)}) svec(B_k^{(j)})^{\mathsf{T}}), \tag{36}$$

where H_0 is $\mathcal{F}_0^{-1} \mathcal{E}_0$, which depends on different choices of symmetrization for search directions. The number of arithmetic operators in (36) is $O(K m_0^2 n_0^4 + n_0^6)$. This completes the proof. □

If we use a generic homogeneous algorithm such as the one in [14], then the number of arithmetic operations required to compute the search directions for (11, 13) is $O(mn^3 + m^2n^2 + m^3)$, where $n = (n_0 + Kn_1)$ and $m = (m_0 + Km_1)$. Setting $m_1 = m_0$ and $n_1 = n_0$ and substituting $n = (1+K)n_0$ and $m = (1+K)m_0$) in $O(mn^3 + m^2n^2 + m^3)$, we have that the complexity of such a generic method of computing the search directions is $O(K^4(m_0n_0^3 + m_0^2n_0^2) + K^3m_0^3)$. This is much larger than the complexity $O(K(m_0^3 + m_0^2n_0^4) + n_0^6)$ obtained for the method in §4 when $K \gg m_0$ and $K \gg n_0$.

It has been shown in [11] that if Problem (11, 13) has a solution, then the KSH method is globally convergent. The algorithm finds an optimal solution or determines that the primal-dual pair has no solution of norm less than a given number in at most $O(n^{1/2}L)$ iterations, where n is the size of the problem and L is the logarithm of the ratio of the initial error and the tolerance. So by using the method in §4 for computing the search direction with KSH symmetrization, the complexity of Algorithm 1 in terms of the total number of arithmetic operations is $O(K^{3/2}(m_0^3n_0^{1/2} + m_0^2n_0^{9/2}) + K^{1/2}n_0^6)$. In comparison, the short- and long-step decomposition algorithms of Ariyawansa and Zhu [2] have complexities of $O(K^{3/2})$ and $O(K^2)$ respectively in terms of the total number of arithmetic operations.

We note that the efficient computation of the Schur computation matrix M_k in (21) is crucial as this is the most expensive step in each iteration where usually 80% of the total CPU time is spent if the algorithm in [15] is used. However, in each iteration, each M_k can be computed independently, and so distributed processing may be used to achieve substantial reductions in computation time.

6 Concluding Remarks

In this paper, we have given a homogeneous self-dual algorithm for Stochastic Semidefinite Programming. Being a homogeneous self-dual algorithm it does not require the user to provide a starting point. Finding a starting point may be difficult for SSDP problems. We have also developed an efficient method for calculating the search directions which can take advantage of parallel and distributed processing. A worst-case bound on the number of arithmetic operations for our algorithm has been obtained. This bound shows that the complexity of our algorithm is similar to that of volumetric decomposition algorithms described in [2].

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5:13–51, 1995.

[2] K. A. Ariyawansa and Y. Zhu. A class of polynomial volumetric barrier decomposition algorithms for stochastic semidefinite programming. *Mathematics of Computation*. In press. (An earlier version of this paper appeared as Technical Report 2006-7, Department of Mathematics, Washington State University, Pullman, Washington 99164-3113 in July 2006.)

[3] K. A. Ariyawansa and Y. Zhu. Stochastic semidefinite programming: A new paradigm for stochastic optimization. *4OR—The Quarterly Journal of the Belgian, French and Italian OR Societies*, 4(3):65–79, 2006. (An earlier version of this paper appeared as Technical Report 2004-10 of the Department of Mathematics, Washington State University, Pullman, WA 99164-3113, in October 2004.)

[4] K. A. Ariyawansa and P. L. Jiang. Polynomial cutting plane algorithms for two-stage stochastic linear programs based on ellipsoids, volumetric centers and analytic centers.Technical Report, Department of Pure and Applied Mathematics, Washington State University, Pullman, WA 99164-3113, 1996.

[5] A. Berkelaar, R. Kouwenberg and S. Zhang. A primal-dual decomposition algorithm for multistage stochastic convex programming. *Math. Program. Ser.A*, 104(1):153–177, 2005.

[6] J. R. Birge. Stochastic programming computation and applications. *INFORMS Journal on Computing*, 9:111–133, 1997.

[7] J. R. Birge and L. Qi. Computing block angular Karmarkar projections with applications to stochastic programming. *Management Science*, 34:1472–1479, 1988.

[8] P. Kall and S. Wallace. *Stochastic Programming.* Wiley, New York, NY, USA, 1994.

[9] S. Mehrotra and M. G. Özevin. Decomposition-Based Interior Point Methods for Two-Stage Stochastic Semidefinite Programming. *SIAM J. of Optimization*, 18(1):206–222, 2007. (An earlier draft of this paper appeared under the title "Two-State Stochastic Semidefinite Programming and Decomposition Based Interior Point Methods: Theory," IEMS Technical Report 2004-16, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, in December 2004.)

[10] Yu. E. Nesterov and A. S. Nemirovski. *Interior Point Polynomial Algorithms in Convex Programming.* SIAM Publications. SIAM, Philadelphia, PA, USA, 1994.

[11] F. Potra and R. Sheng. On homogeneous interior-point algorithms for semidefinite programming. *Optim. Methods Softw.*, 9:161–184, 1998.

[12] B. Straziky. Some results concerning an algorithm for the discrete recourse problem. In *Stochastic Programming*, ed. M. Dempster, Academic Press, London. 263–274, 1980.

[13] M. J. Todd. Semidefinite Optimization. *ACTA Numerica*, 10:515–560, 2001.

[14] M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov-Todd direction in semidefinite programming. *SIAM J. Optim.*, 8:769–796, 1998.

[15] K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 1.3. Interior point methods. *Optim. Methods Softw.*, 11/12:545–581, 1999.

[16] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38:49–95, 1996.

[17] R. VanSlyke and R. J-B. Wets. L-shaped linear programs applications to optimal control and stochastic programming. *SIAM J. App. Math.*, 17:638–663, 1969.

[18] R. Wets. Programming under uncertainty: The equivalent convex program. *SIAM J. App. Math.*, 14:89–105, 1966.

[19] R. Wets. Programming under uncertainty: The solution set. *SIAM J. App. Math.*, 14:1143–1151, 1966.

[20] G. Y. Zhao. A log-barrier method with Benders decomposition for solving two-stage stochastic linear programs. *Math. Prog.*, 90:507–536, 2001.

[21] Y. Zhu and K. A. Ariyawansa. A preliminary set of applications leading to stochastic semidefinite programs and chance-constrained semidefinite programs. *Applied Mathematical Modelling*, 35:2425–2442, 2011. (An earlier version of this paper appeared as Technical Report 2006-8, Department of Mathematics, Washington State University, Pullman, WA, WA 99164-3113 in 2006.)